

All common knowledge about
**HEX-EDITING
SAVED GAMES**
and more

Last updated May 22st, 2002

By various authors, mentioned where appropriate.
This text may be copied freely without permission.
All information has been edited and combined by
Allard H.S. Höfelt (<mailto:ahofelt@hotmail.com>).

This file is written specifically for use with Fantastic Worlds (v2.7.81), an add-on disk for Civilization II. Most information, apart from some offsets, will be about the same with files saved under other versions. Especially with Civilization: Test of Time the offsets are greatly different, but still the structure is still more or less the same.

Contents of this document:

1. LAYOUT OF A SAVED GAME.....	2
2. START OF SAVED GAME FILE.....	3
3. WONDERS.....	5
4. TRIBES.....	6
5. TECHNOLOGIES & MONEY.....	7
6. MAPS.....	9
6.1 SAV files.....	9
6.2 MP files.....	11
7. UNITS INFORMATION.....	13
8. CITIES INFORMATION.....	14
9. BUILT CITIES INFORMATION & MORE.....	16
A. HOW TO HEX-EDIT.....	17
B. VERSION HISTORY.....	18
C. CREDITS.....	19

1. LAYOUT OF A SAVED GAME

Hex offsets are given here, for as long as the values are fixed.

From the map data block 1 onwards, the saved game values are no longer fixed.

The offsets here are for FW, but the structure remains the same in all versions of Civ2.

The offsets are given in **decimals**.

00 - 47	toggled options, various player settings
48 - 265	unidentified or empty
266 - 321	Wonders location (with city ID)
322 - 583	unidentified or empty
584 - 2277	7 blocks of each F2 bytes long with city style information, leader and tribe names for all 7 civs
2278 - 13701	8 blocks of each 593 (in hex) bytes long with various information, like money and techs (a lot still unidentified) for all 8 civs (incl. barbarians).
13702 - 13716	map header information (7 bytes)
...	map data block 1 (7 * height * width bytes)
...	map data block 2 (6 * width * height bytes)
...	map data block 3 (2 * quarterwidth * quarterheight bytes)
...	1024 unused bytes
...	unit information (26 * number of units bytes)
...	city information (84 * number of cities bytes)
...	one empty buffer byte
...	3 bytes for all 21 civs containing info as which city in CITIES.TXT to pick
...	unidentified, and other info such as where the screen is centred or the cursor is located.
...	a bit of text on how the scenario is called, and probably a bit of history (who's wiped out when, etc.).
...	event stuff (starts with EVNT, easy to find).

(In MGE a unit entry is 32 bytes long versus 26 for earlier versions and city entry is 88 bytes long versus 84 for earlier versions.)

2. START OF SAVED GAME FILE

(source Allard Höfelt)

(Xin Yu found out about byte 2F toggling civ to be played)

(Captain Nemo found out the civs that are in play in byte 2E)

0-8	"CIVILIZE" and one empty byte (00)
10	27 : Conflicts in Civilization or lower
	28 : Fantastic Worlds (2.7.81)
	2C : Multiplayer Gold
	31 : Test of Time
12	1... toggles: bloodlust
	...1 ... simplified combat
13	1... flat/round world
	1 don't restart if eliminated
14	1... move units without mouse
	.1... enter closes city screen
	..1... map grid
	...1 ... sound effects
	... 1... music
15	1... cheat menu (also see 20)
	.1... always wait at end of turn
	..1... autosave each turn
	...1 ... show enemy moves
	... 1... no pause after enemy moves
	1.. fast piece slide
	1. instant advice
	1 tutorial help
16	..1... animated heralds
	...1 ... High Council
	... 1... civilopedia for advances
	1.. throne room graphics
	1. diplomacy screen graphics
	1 wonder movies
20	...1 ... cheat penalty/warning
	.1... scenario file (no effect really)
	1... toggle scenario flag (thanks to Kull)
22	1... announce "we love the king day"
	.1... warn when food dangerously low
	..1... announce cities in disorder
	...1 ... announce order restored in city
	... 1... show non-combat units built
	1.. show invalid build instructions
	1. warn when city growth halted
	1 show city improvements built
23	1.. zoom to city not default action
	1. warn when new pollution occurs
	1 warn when changing production will cost shields
28-29	number of turns passed
30-31	number of turns passed to calculate game year in pop-ups, status bar, etc.
34-35	specifies which unit is selected at start of the game
	(you can find the unit id number by playing another civ, revealing the map and right clicking on the unit. On the right, the id is shown between parentheses)
39	changes which human player is used
40	player's map which is used
41	player's civilization number used
42	changes with map used. Sometimes FF
43	map revealed or not
44	difficulty level [deity=5, emperor=4, etc.]
45	barbarian activity
46	enumeration of all civs still in play [binary]
	eg. 1001 1101 means: barbarians alive, civ 3 alive, civ 4,5 and 7 alive.

- 47 human player played (can be more than one) **[binary]**
 toggling this byte is great and allows Hotseat mode in FW!!!
 eg. 0110 0100 means: human is playing as civ 1, 2 and 5.
- 50 amount of current pollution. 7F is maximum, and will certainly cause global temperature rising at the end of the turn. A value of 80 till FF is a negative amount (still shows the icon for global rising), but it will be reset to 00 at the end of the turn.
- 51 amount of times that a global temperature rising has been happening this far in the game. 7F is maximum and will make the whole world a big swamp when the next global temperature rising occurs. 80 or higher will prevent any global rising from happening at all.
 At the moment when terrain should be changed, nothing happens, and byte 32 is restored to 00, though a message still pops up. Byte 33 will remain at what it was, so it is not reset.
 This is a fool-proof way to switch off pollution without the restrictions of the normal cheat-menu switching off of pollution. The message "global warming occurs/will occur soon" of course will need to be changed.
- 56 number of turns of peace (counts only after 200th turn)
- 58.59 Total number of units in units list. Probably not necessarily the exact amount of units, as some spaces in the list can be empty, after the unit is destroyed. Lowering this will remove the most recently built units. Making it higher will produce very strange results, because the game will read the saved game file incorrectly.
- 60.61 Total number of cities in city list. Probably not necessarily the exact amount of cities, as some spaces in the list can be empty, after the city is destroyed. Lowering this will remove the most recently built cities. Making it higher will produce very strange results, because the game will read the saved game file incorrectly.

3. WONDERS

(source Allard Höfelt)

Near the start of the saved game, at offset 266 until 321, is where all information about wonders is stored. All 28 wonders have 2 bytes of information for them, so in total the block for all wonders is 56 bytes long.

If the 2 bytes for a wonder are

- FF FF : the wonder has not yet been built.
- EF FF : the wonder is destroyed (city where it stood is not stored).
- otherwise the number is the city ID number. 00 00 the very first city built, etc.

4. TRIBES

(source Allard Höfelt)

The information is stored in 7 blocks (**not** for barbarians), each 242 bytes long.

The blocks start and end at:

584-825	1 st block	
826-1067	2 nd "	(numbers given in decimals)
1068-1309	3 rd "	
1310-1551	4 th "	
1552-1793	5 th "	
1794-2035	6 th "	
2036-2278	7 th "	

These 7 blocks are all filled with 00's except for a few names and numbers.

Counting the city style as the first byte (also the first byte in the block), the blocks are divided as:

1 st byte	city style (can be any style, also industrial and modern or even higher (strange effects), without the required advances. Just fill in 00, 01, 02....)
3-25	Leader name. Empty letters "00".
27-49	Tribe name.
51-73	Tribe name adjective.
75-97	Leader title 1 (Anarchy)
99-121	Leader title 2 (Despotism)
123-145	Leader title 3 (Monarchy)
147-169	Leader title 4 (Communism)
171-193	Leader title 5 (Fundamentalism)
195-217	Leader title 6 (Republic)
219-241	Leader title 7 (Democracy)

5. TECHNOLOGIES & MONEY

(source Allard Höfelt)

Byte 2287 until 13701 is divided in 8 similar blocks of each 1427 bytes. Each of these blocks contains information specific for each civ, such as technologies and money. A lot more in these blocks has as yet not been identified.

The blocks all start at these offset bytes:

Block of	0 th civ: 2278 – 3705 (barbarians)	
"	1 st civ: 3706 – 5133	
"	2 nd civ: 5134 – 6561	
"	3 rd civ: 6562 – 7989	(offsets given in decimals)
"	4 th civ: 7990 – 9417	
"	5 th civ: 9418 – 10845	
"	6 th civ: 10846 – 12273	
"	7 th civ: 12274 – 13701	

Counted from the start of each block as byte 1:

byte 2 Gender. Male=00 Female=02

byte 3-4 Money. The barbarians' money is also changeable.

byte 7 What civ number (as determined in RULES.TXT) is in play here. Can be a civ of the same colour for more than one civ in play, or even the same civ. You can for example make 7 players Romans, with the same colour, portrait, flag, etc.
Discovered by Andrew Livings.

byte 9 Research progress. Also updates the colour of research indicator in status bar on the right.

byte 11 Tech which is being researched. FF if cleared or no goal.

byte 21 Percentages to tax, science and luxuries.

byte 22 Government. Valid values are 00-07 (though 07 does not normally exist). A value of 08 or higher causes the game to crash. 00 is of course Anarchy, 01 is Despotism, etc.
You can give the barbarians a Democratic (06) government to prevent their cities being subverted.

byte 31 Reputation

byte 37-64 Treaties. 4 bytes for the treaties with all 7 civs (including their own civ).

1st byte:	1 - contact
	1. - cease fire
	1.. - peace
	 1... - alliance (with peace always)
	...1 - vendetta
	1... - embassy
2nd byte:	..1. - war
	1.. - ????
3rd byte:	..1. - ????
4th byte:	????

then the 1st byte for the next civ, for all 7 civs.

NOTE: You can also combine these values, for example to create a war and an alliance at the same time. In this example, your units will be repaired when attacking enemy cities, but in the diplomacy screens the civ will talk with you as if you were at war. More similar strange effects can occur.

byte 66-72 Attitudes. Byte 42 is attitude to 1st player, 43 is to 2nd player, etc.

byte 89-100 Technologies. B (11) bytes long.

To explain how the bytes work, an example of the first technology byte:

```
.... ..1 : 1st technology
.... ..1. : 2nd  "
.... ..1.. : 3rd  "
.... 1... : 4th  "
...1 .... : 5th  "
..1. .... : 6th  "
1... .... : 7th  "
```

In the next byte are the 8th-14th technology. Note that for some reason the .1.. is left out. This is not the case in each of these bytes, though. I don't know why. Though it's fun to know how it works, it is much much easier to set techs using the cheat menu.

NOTE: This is very useful, however, for giving the Barbarians certain technologies. You might want to give them Guerrilla Warfare, for instance.

NOTE: All techs are: FF FF FF FF FF FF FF FF FF FF FD. (Excluding user defined techs. Adding those makes it FF instead of FD).

byte 103-104 military demographics value.

Byte 105 city count (found by Carl Fritz). Shows up in MGE when you start a multiplayer game. Otherwise no apparent effect.

byte 997-1010 last contact with civs. First two to 1st civ, 3rd and 4th for 2nd civ, etc. FF FF if not available, 01 00 for 1, etc.
note: It may be that the FF FF in byte 3E3 and 3E4 are for last contact with barbarians, but that is not verifiable.

6. MAPS

(source Jorrit Vermeiren, Carl Fritz, James Dustin Reichwein, Angelo Scotto, Harlan Thompson)

If you have an Hex Editor which has a graphical mode, this is very handy for especially the maps part. Orient the hex editor to start viewing at the beginning of map block 3 and make the width of the display the same as the 'locator map X dimension', and then you can see two little maps!

The following is all taken literally from Jorrit's guide on map hex editing.

Civilization II Map Structures

This file describes the way the map is encoded in CivilizationII maps and savegames. This data is acquired using the Fantastic Worlds add-on (Civ2 version 2.7.8I). Savegame structures from later Civ2 versions (notably CivilizationII: Test of Time) may vary significantly.

In this document I assume the reader has some basic knowledge of hexadecimal and binary numbers, as well as some familiarity with a few essential terms like unsigned short integer or offset. This shouldn't be much of a problem for anyone with a bit of programming or hexediting experience.

Most of the times, I have marked hexadecimals with the prefix 0x. As can be expected with a Windows-based application, data types are stored in Little Endian fashion (e.g. the short integer 10000 is 0x2710 in hexadecimals, but is stored as 0x1027).

Credits go to [James Dustin Reichwein](#), who discovered a few important details of the 6th byte in the 2nd block, [Angelo Scotto](#), who pointed out to me that a map wasn't just represented by block 2 in savegames and scenarios, and [Harlan Thompson](#), who directed me towards finding the no-resource bit in the terrain codes.

[Jorrit Vermeiren](#)

.MP Map Structure

Map Header (49 unsigned short integers, offset 0x00000000)

offset	value
0x00000000	2 * width (so in a 50 x 80 map this value would be 100)
0x00000002	height
0x00000004	surface (width * height)
0x00000006	shape (0 is round, 1 is flat)
0x00000008	seed (1 is random, 2-64 are valid seed patterns, anything above 64 maps to one of these values. I.e 65 = 1, 66 = 2 etc.)
0x0000000A	width / 2 (rounded up)
0x0000000C	height / 4 (rounded up)
0x0000000E	X coordinates of all 21 civilization starting points, in the same order as in the RULES.TXT. When a civilization has no starting point, the value will be 0xFFFF NOTE: Both X and Y coordinates are in the Civ2 coordinate system, NOT the Map Editor coordinate system!
0x00000038	Y coordinates of all 21 civilization starting points, in the same order as in the RULES.TXT.

Map Data (6 * surface bytes, offset 0x00000062)

	value (hexadecimal)
1	Terrain code
2	00
3	00
4	00
5	Terrain code

6	F0
---	----

Each square is encoded by the above mentioned 6 bytes. The order of the squares in the file are in regular reading order (i.e. left to right, top to bottom).

.SAV Map Structure

The map data in a CivilizationII .SAV file consists of four different blocks:

- The map header
- Civilization-specific tile information
- General map information
- ???

Map Header (7 unsigned short integers, offset 0x00003586)

offset	value
0x00003586	2 * width
0x00003588	height
0x0000358A	surface (width * height)
0x0000358C	shape
0x0000358E	seed
0x00003590	width / 2 (rounded up)
0x00003592	height / 4 (rounded up)

Note that the above mentioned numbers are exactly the same as the the first seven values in a map file.

Map Data BLOCK 1 (7 * surface bytes, offset 0x00003594)

This block is actually 7 blocks (with 1 byte per square) behind each other. Each of these blocks represents some data for each of the civilizations. So, the first block contains information about the white civ, the second block contains information about the green civ etc.

Each of these blocks encode the known tile improvements for the civilization in question. [The second byte in BLOCK 2](#) uses these same codes, except that then all actual tile improvements of all civilizations are mentioned:

hexadecimal	binary	tile improvement
00	00000000	"Nothing", i.e. the square in question contains no (known) units or tile improvements.
01	1	Unit present
02	1.	City present
04	1..	Irrigation
08	1...	Mining
10	...1....	Road
20	..1.....	Railroad
40	.1.....	Fortress
80	1.....	Pollution

Farmland is irrigation plus mining (0x0C). Railroad is actually 0x30, a combination of road and the "road upgrade" 0x20. An airbase is a combination of the fortress and city present codes: 0x42. In CivilizationII: Test of Time, transport sites are marked with 0x82, an otherwise impossible combination of the pollution and "city present" codes.

Generally speaking, the codes are cumulative, that is, completely in parallel with what you'd expect in Civ2. E.g. road and pollution can co-exist, but irrigation and farmland cannot, since irrigation is part of the farmland code.

All cities, including unknown cities, are marked. Undiscovered (see [5th byte in BLOCK 2](#)) cities are generally (not always) assumed to be ungarrisoned (i.e. code is 0x02, not 0x03). A city that has never been discovered and is already destroyed will remain on a civilization's map, until

the area is "re-explored". Enemy cities or units that are lost out of sight, will remain marked on their last seen location (?), but again only as long as no news about the area is gathered.

Map Data BLOCK 2 (6 * surface bytes)

This block contains only one map, but each square is now represented by a sequence of 6 bytes:

1. The first byte determines the terraintype. The terrain codes are exactly the same as [the codes used in the .MP format](#).
2. The second byte determines the tile improvements. The tile improvement codes are the same as [the codes used in the first block](#).
3. This byte is an indicator for the city radii. Each civilization color has its own city radius code:
White 0x20
Green 0x40
Blue 0x60
Yellow 0x80
Cyan 0xA0
Orange 0xC0
Purple 0xE0

When two cities of two different civilizations share a part of their city radius the squares in question will be given to one of those cities, not both. It's very likely there is some sort of an algorithm to determine how squares are divided, perhaps the tiles are simply assigned randomly.

Barbarian cities and squares that aren't part of a city radius will have the code 00.

4. The 4th byte is a land/sea body counter. This simply gives every different body of land and every body of water a separate number. Counting starts at the top-left corner and proceeds from left to right and from top to bottom. Water bodies of less than 9 squares always have number 63, but do count towards the total. Both the land and water counters start counting at one.

NOTE: To see what I mean, right-click on a square in CivilizationII. In the Status Window you'll see something like this:

Loc: (12, 34) 5

12 is the X-coordinate of that square

34 is its Y-coordinate

5 is this "body counter".

5. The 5th byte of the 6 bytes represents "visibility", indicating which of the civilizations have explored that particular square:

hexadecimal	binary	civilization
00	00000000	No-one has discovered this square.
01	1	Red (Barbarians)
02	1.	White
04	1..	Green
08	1...	Blue
10	...1....	Yellow
20	..1.....	Cyan
40	.1.....	Orange
80	1.....	Purple

6. The 6th byte actually encodes two things. The first 4 bits concern tile ownership. An indicator for this tile ownership is the color of an airbase. A funny effect that can be created with faulty ownership values is that a civilization attacks its own unit. The used codes are as follows:

hexadecimal	owner
0.	Red (Barbarians)
1.	White
2.	Green
3.	Blue

4.	Yellow
5.	Cyan
6.	Orange
7.	Purple
8. - E.	Illegal values
F.	None (Barbarians?)

The last 4 bits concern tile fertility. Values for fertility range from 0 to 15 (which is 0 to F in hexadecimals), 0 being completely infertile, 15 the most fertile. How the exact algorithm works is not clear to me, but I do have a few valuable clues: CivilizationII only assigns fertility values to the Grassland and Plains terrains, all other terrains initially have a fertility value of 0. This explains why the AI only builds cities on these terrain types. Fertility values of terrains other than Grassland or Plains can, however, increase because of tile improvements. The AI will not build cities on terrains with a fertility value below 7. Actual values are influenced by production, food and trade as defined in the RULES.TXT as well as tile improvements, such as roads and irrigation. Fertility values within a city radius are decreased by a certain number, effectively preventing the AI from building cities within any existing city radius.

Fertility seems to be much more complicated than just a square by square calculation, though. Data from one game showed that fertility values ranged from 0 to 7 at the first turn of the game. The next turn, when 4 capital cities had been built, all non-zero fertility values had increased by 2, while the city radius squares now had a value of 2. At the end of another game on the same map, city radius squares had values of 6, while other fertile squares now had values ranging from 9 to 15. I am inclined to say that fertility values also depend on the total amount of fertile squares that are still available.

Map Data BLOCK 3 (2 * Qwidth * Qheight bytes)

The size of this block equals twice the multiplication of the 6th and 7th values in the [map header](#). With the Qwidth and Qheight names I refer to those values in the map header. This block consists of two parts of equal size. The pattern in these two parts roughly equals the map shape, only making a distinction between land and ocean. Because both width and height of these parts are only a fraction of the map's measurements there is no one-to-one relationship between the bits or bytes in this section and the squares on the map. As of yet, I have no idea what the purpose of this block is. The block never changes during a game and changing it doesn't seem to influence a game whatsoever.

After BLOCK 3...

The third map data block is followed by 1024 bytes. I have no idea what that represents, but it doesn't seem to have anything to do with the map. After these mysterious bytes follows the unit information section.

Terrain Codes

hexadecimal	binary	terrain
00	00000000	Desert
01	00000001	Plains
02	00000010	Grassland
03	00000011	Forest
04	00000100	Hills
05	00000101	Mountains
06	00000110	Tundra
07	00000111	Glacier
08	00001000	Swamp
09	00001001	Jungle
0A	00001010	Ocean
40	0100....	No Resource
80	1000....	River

[Jorrit Vermeiren](#)

Last updated: 25 March 2002

7. UNITS INFORMATION

(source Allard Höfelt)

Located right before the city information. It may be handy to set you hex editor to show 13/26 bytes per row, so that a unit is exactly 2/1 rows under the one before him. The units are numbered (right click on a unit and you'll see its ID number) according to the way they're ordered in here.

Find specific unit by toggling eg. veteran status. Alternatively, search for the location of the unit by searching the x and y coordinates. If your unit is located at (88,46) look for the hex value 58 00 2E 00.

- 01-02 horizontal coordinates of unit
(NOTE: the unit is invisible if moved, unless the terrain it now occupies has the "contains unit" flag toggled on. See terrains)
- 03-04 vertical coordinates of unit (same NOTE as with horizontal)
- 06 1... adds a grey star to the shield, no apparent function
..1. toggles veteran
- 07 unit type (counted in rules.txt, 00=settler, 01=engineer, etc.)
- 08 owning civilization
- 09 movement of unit (don't know how it works exactly)
- 11 number of hitpoints lost
- 12 ?? does change
- 14 caravan commodity
(00 hides, 01 wool, ..., 0A uranium, F0 food supply, rest makes no sense)
- 16 orders:
 - 01 : fortify (not yet fortified, but ordered to do so)
 - 02 : fortified
 - 03 : sentry
 - 04 : build fortress
 - 05 : builds road/railroad
 - 06 : builds irrigation
 - 07 : builds mine
 - 08 : transform terrain
 - 09 : cleans up pollution
 - 0A : builds airbase
 - 0B : go to (see number 19-20 and 21-22)
 - FF : no orders (can also fill in any other number for no orders, but computer uses FF)
- 17 home city with number is city number (FF is none)
(You can find out the city number with for example "find city". If all cities that are on the map are listed, you can simply count them to find the number)
- 19-20 horizontal coordinate of "go-to" command
- 21-22 vertical coordinate of "go-to" command
- 23-24 link to ?id?'s of other units to be drawn in the same square on top of the unit
(CAUTION: can produce strange effects)
- 25-26 link to ?id?'s of other units to be drawn in the same square under the unit
(CAUTION: can produce strange effects)

...at location 27, the next unit starts all over again.

In MGE, the next unit start at location 33 (so one unit consists of 32 bytes instead of 26).

JP Notes: In ToT, it's 40 bytes per unit. The first two pairs of bytes are the map coordinates. Byte 08 is the veteran flag, byte 09, the unit type, byte 10, the owner, and so on (following the order given above).

There is a problem with byte number 18 (orders) when using Scotto's conversion utility. In Civ2, a value of 0B indicates the "go to" order. In ToT, this value flags the "build transporter" order and 0C is used for the "go to" order. Therefore, when converting scenarios, these bytes need to be manually changed from 0B to 0C, otherwise units will begin building transporters in the first turn of the game.

8. CITIES INFORMATION

(source Andrew Livings and Xin Yu)

The easiest way to locate this part of the saved game is to make a search for the name of a city. They are located near the end of the save game.
Keep in mind that the city name itself is on the 33th hex numbers pair of the information block about that city.

You can move the city to another location by changing the vertical and horizontal coordinate of the city. Keep in mind though that for the city to show up, the square where the city will stand needs to have the "city present" flag enabled. See 6.1.2 (map blocks) for more information about this.

You can have multiple cities on one square, you can have cities adjacent to each other, you can have cities in the sea. Everything is possible.

- 1-2 vertical coordinates of city
- 3-4 horizontal coordinates of city
- 5 1... can build coastal improvements
...1 ... denotes auto-build. Under whose rule is in byte 8
... 1... denotes stolen tech
... .1.. denotes improvement sold
... ..1. denotes "we love the king day"
... ...1 denotes civil disorder
- 7 ..1. can build ships (only if coastal flag is checked)
- 8 ...1 ... denotes objective x3
... .1.. denotes objective x1 (you can also set both the x3 and x1 objective)
... ..1. denotes auto-build under domestic advisor
... ...1 denotes auto-build under military advisor
... ..00 (both military and domestic flags to 0) to turn
on both at same time. Note that these also need byte 5
the auto-build flag to be active.
- 9 city owner [00-07 (don't use >7)]
- 10 city size (negative sizes cultivate no squares but produce food)
- 11 who originally built the city [00-07]
- 15-21 working city production squares??
- 23-26 specialists, 4 hex positions representing 16 base 4 numbers representing specialists
0=none, 1=elvis, 2=tax, 3=sci.
example: 06000000, first hex 06=1*4+2: 1 taxman and 1 elvis
- 27-28 food in food box (FF=famine)
- 29-30 shields in shield box
- 31-32 net trade (not including those from trade routes)
- 33-47 city name buffer [15 characters, end with a 00]
- 48 00 (because city name cannot exceed 15 characters)
- 49 workers in inner circle (8 squares next to city square) [ff=all worked]
- 50 workers on 8 of the outer circle
- 51 workers on 4 of the outer circle
[first digit is always 1, the second digit represents the 4 working positions]
- 52 number of specialists times 4 [eg. 08=2 specialists]
- 53-57 city improvements:
1... is 7th improvement
.1.. is 6th improvement
..1. is 5th improvement
...1 ... is 4th improvement
.... 1... is 3rd improvement
.... .1.. is 2nd improvement
.... ..1. is 1st improvement
- 58 item in production (can be used to force production)
[units 00-3f]
(improvements are inversed [FF=1st FE=2nd etc])
- 59 number of active trade routes [decimal, can also be >03]
- 60-62 = 1st, 2nd and 3rd trade commodities available
- [00-0F (are 1st-15th in rules.txt)]
- supplied trade goods, eg (Coal), are inverse values.

eg., commodity 13 is already supplied, the value is F3 (-13)
- if in parentheses use FF compliment (can force trade commodity
eg., 0F=Uranium, F0 = (Uranium), which is ff-0f

63-65 = 1st, 2nd and 3rd trade commodities demanded

66-68 = 1st, 2nd and 3rd trade commodities in route

69-74 = 1st, 2nd and 3rd trade route partner city number (city number is 85th-86th position)

75-76 = science

77-78 = tax

79-80 = number of trade icons (including trade routes)

81 = total food production

82 = total shield production

83 = number of happy citizen

84 = number of unhappy citizen (double unhappy count twice??)

...at location 85, the next city starts all over again.

In MGE, the next city start at location 89 (so one city consists of 88 bytes instead of 84).

In ToT, the next city start at location 93 (so one city consists of 92 bytes instead of 84).

The first two pairs of bytes are the map coordinates. Byte 09 is the build ships flag, byte 10, the objective flag, byte 11, the owner, and so on (following the order given above).

9. BUILT CITIES INFORMATION & MORE

(source Allard Höfelt)

After the 84th byte of the very last city, one byte is empty (00).

Let us call the byte thereafter number 1.

Byte 1-3 are for the Romans, 4-6 for Babylonians, 7-9 for Germans, ... for all 21 civs.

Of these three bytes, the first byte is the amount of cities that this civ built so far. This value is used to pick **which city name to use in CITIES.TXT**. The second and third bytes for each civ are not identified, and appear to sometimes be 00, sometimes 01.

If we continue this numbering from the byte we called 1, we can also locate some more values:

- 63th-66th bytes are the coordinates of the cursor.

A. HOW TO HEX-EDIT

There are a few things I would like to note about Hex Editing files:

- **Integers are almost always stored in reverse. If 300 turns have passed (300 is 01 2C in hex), you should type it in as 2C 01 instead of 01 2C.**
- Binary values are often used as “flags”, which can be toggled on or off. The way I have noted these in this document is eg ..1., which means that “.” can be either 0 or 1, but has no influence on the actual flag which is meant. 1, of course, must be 1 for the flag to work.
- I have tried to give most numbers in Decimals for better understanding. However, some important offsets are still given in hex. They are either obvious to find or are noted.

There are several Hex Editors available. Most are downloadable from the Internet.

I recommend two editors:

- Hex Workshop, <http://www.bpsoft.com> with many good options, including a very useful hexadecimal-decimal converter and a hexadecimal calculator. The program is shareware, but there are cracks for it everywhere. Just type in the name and version number in <http://astalavista.box.sk> and download a crack or serial to remove the shareware restrictions.
- A.X.E., <http://www.kahei.com>. A free editor, which has a very useful graphical viewing option. Some people swear by this program, but I myself prefer Hex Workshop. *NOTE: This program does not seem to be downloadable anymore from this site, but perhaps you can still download it from other sources.*

Another major use for hex-editing not elaborated on in here is for extracting GIF images from DLL's. The program usually used for this and especially created for it is GifX by Rune Berge. However there are other resource extracting programs not specifically for CivII that do a similar job, and can even be better than GifX.

- **GifX** by Rune Berge. Downloadable at every good civ2 scenario site.
- **Resource Hacker** or **ResHacker** by Angus Johnson. This program allows editing of most 32bit resource files, among others the Civ2.exe file (to extract some funny things) and all civ2 DLL files. However, it only allows editing of 32bit files. Civ2 was written in 16bit (Windows 3.1), so these cannot be edited. Multiplayer Gold (MGE) was rewritten as 32bit, and these files can be edited. The program is **free** and can be downloaded here: <http://rpi.net.au/~ajohnson/resourcehacker/>.
- **Exescope** by Toshifumi Yamamoto. This program is NOT free, but cracks or serials can be found at <http://astalavista.box.sk>. This program essentially does the same as ResHacker, but also allows the editing of 16bit files, so all versions of civ2. The main advantage with editing DLL's over GifX is that this program allows file sizes to be changed, so you don't have to worry anymore about your edited picture being larger in size than the original. Download it here: <http://www.vector.co.jp/authors/VA003525/> (and click on *English*).

B. VERSION HISTORY

- | | |
|---|---|
| 1.6 (22 th May 2002) | Lots of additions, changes and extra info. More tips on programs for resource editing in DLL and EXE files. Replaced the whole chapter on map editing with Jorrit's guide on map editing. |
| 1.5 (27 th August 2000) | Some minor additions. |
| 1.4 (13 th August 2000) | Added Jorrit Vermeiren's part about maps. Reformatted the whole text to Word. Added leader titles, total number of units and cities in game, cities to pick in CITIES.TXT. |
| 1.3 (20 th June 2000) | Added the Wonders section. |
| 1.2 (26 th May 2000) | Added some unimportant info, pollution info, more info about Technologies and Money and some zoom factors. |
| 1.1 (15 th May 2000) | Added cursor coordinates, the Save Game layout some game versions, Tribes info, Technologies and Money info. |
| 1.0 (5 th May 2000) | Original version. |

C. CREDITS

Sources are mentioned where appropriate. The people who contributed to this document are in no particular order:

- Allard Höfelt (the writer of this document)
- Jorrit Vermeiren
- Andrew Livings
- Carl Fritz (Gothmog)
- Xin Yu
- Captain Nemo

Thank you to all people who have contributed by asking valuable questions and by this way finding more out.